

**CLAIMS**

1. A method for language verification of a Java card CAP file created from an original Java code file, comprising:
  - a) a conversion step for converting said Java card CAP file into a corresponding converted Java code file that is semantically identical to said Java card CAP file; and
  - b) a language-verification step for verifying said converted Java code file for compliance with Java language specifications.
2. A method for language verification of a Java card CAP file according to Claim 1, wherein said conversion step further comprises:
  - a preconversion substep for converting Java card IDs contained in said Java card CAP file into symbolic names, and for converting said Java card CAP file into a standard Java format, to obtain a preconverted file; and
  - a mapping substep for replacing in said preconverted file externally defined names with original names by using a mapping scheme between Java names and tokenized identifiers, to obtain the converted Java code file for said language-verification step.
3. A method for language verification of a Java card CAP file according to Claim 2, wherein said mapping substep is performed using a referenced Java export file which is available as a result of creating said Java card CAP file from said original Java code file.

4. A method for language verification of a Java card CAP file according to Claim 1, further comprising:

c) a signature step for creating, after verification of said converted Java code file in said language verification step, a second cryptographic signature file.

5 5. A method for language verification of a Java card CAP file according to Claim 4, further comprising:

d) a loading step for loading the second cryptographic signature file to a storage device together with the Java card CAP file.

6. A method for language verification of a Java card CAP file according to Claim 4, wherein the second cryptographic signature file is cryptographically verifiable, said method further comprising:

e) an executing step for executing said Java card CAP file upon a positive cryptographic verification.

7. A method for language verification of a reduced file derived from an original file, the reduced file conserving original semantics, said method comprising:

a) a conversion step for converting said reduced file into a corresponding converted file that is semantically identical to said reduced file; and

b) a language-verification step for verifying said converted file.

8. A method for language verification of a reduced file according to Claim 7, wherein said conversion step further comprises:

a preconversion substep for converting IDs contained in said reduced file into symbolic names, and for converting said reduced file into a standard format, to obtain a preconverted file; and

a mapping substep for replacing in said preconverted file externally defined names with original names by using a mapping scheme, to obtain the converted file for use in said language-verification step.

9. A method for language verification of a reduced file according to Claim 8, wherein said mapping substep is performed using a referenced difference file which is available as a result of deriving said reduced file from said original file.

10. A computer program product comprising program code means for language verifying a Java card CAP file, comprising:

a) first processes for converting said Java card CAP file into a corresponding converted Java code file that is semantically identical to said Java card CAP file; and

b) second processes for verifying said converted Java code file for compliance with Java language specifications.

11. A computer program product for language verifying a Java card CAP file according to Claim 10, wherein said second processes further comprises:

first subprocesses for converting Java card IDs contained in said Java card CAP file into symbolic names, and for converting said Java card CAP file into a standard Java format, to obtain a preconverted file; and

for replacing in said preconverted file externally defined names with original names by using a mapping scheme between Java names and tokenized identifiers, to obtain the converted Java code file.

5 12. A Java card CAP file language verifier for verifying a Java card CAP file that has been derived from an original Java code file, said Java card CAP file including original Java semantics of said original Java card file, comprising:

a converter for converting said Java card CAP file into a corresponding converted Java code file that is semantically identical to said Java card CAP file; and

10 a language verifier for verifying said converted Java code file upon its compliance with a Java language specification.

13. A Java card CAP file language verifier according to Claim 12, wherein said converter further comprises:

15 a preconverter for converting Java card IDs contained in said Java card CAP file into symbolic names, and for converting said Java card CAP file into a standard Java format, to obtain a preconverted file; and

a mapper for replacing in said preconverted file externally defined names with original names under use of a mapping scheme, to obtain the converted Java code file.

20 14. A Java card CAP file language verifier according to Claim 13, wherein the mapper comprises an input for receiving a referenced Java export file created when a referenced Java card CAP file was converted from its corresponding original Java code file.

15. A Java card CAP file language verifier, according to Claim 12, further comprising signature generator for generating a second cryptographic signature file.
16. A Java card CAP file language verifier, according to Claim 15, further comprising a means for loading the second cryptographic signature file and the Java card CAP file to a storage device.
17. A reduced file language verifier for verifying a reduced file that has been converted from an original file, the reduced file maintaining original semantics of the original file, comprising:
- a converter for converting said reduced file into a corresponding converted file that is semantically identical to said reduced file;
  - means for determining whether said reduced file complies with a predetermined language specification; and
  - a language verifier for verifying said converted file upon compliance with the predetermined language specification.
18. A reduced file language verifier according to Claim 17, wherein said converter further comprises:
- a preconverter for converting IDs contained in said reduced file into symbolic names and for converting said reduced file into a standard format, to obtain a preconverted file; and
  - a mapper for replacing in said preconverted file externally defined names with original names under use of a mapping scheme, to obtain the converted file.

19. A reduced file language verifier according to Claim 18, wherein said mapper comprises an input for a referenced difference file which is available as a result from a conversion in which a referenced reduced file has been converted from its original file.

\* \* \*

[illegible]